# Benchmarking Hadoop Performance in the Cloud - An in Depth Study of Resource Management and Energy Consumption

Aymen Jlassi, Patrick Martineau

# Benchmarking Hadoop performance in the Cloud
## *An in depth study of resource management and energy consumption*

Aymen Jlassi[1 2] and Patrick Martineau[1]
[1] *Université François-Rabelais de Tours, CNRS, LI EA 6300, OC ERL CNRS 6305, Tours, France*
[2] *Groupe Cyrès, Tours, France*
*aymen.jlassi@etu.univ-tours.fr, patrick.martineau@univ-tours.fr*

Keywords: Cloud Computing, Virtualization, Green Consumption,Docker Container, Hadoop, Resources Consumption.

Abstract: Virtual technologies have proven their capabilities to ensure good performance in the context of high performance computing (HPC). During the last decade, the big data tools have been emerging, they have their own needs in performance and infrastructure. Having a wide breadth of experience in the HPC domain, the experts can evaluate the infrastructures used to run big data tools easily. The outcome of this paper is the evaluation of two technologies of virtualization in the context of big data tools. We **compare the performance and the energy consumption of two technologies of virtualization (Docker containers and VMware) and benchmark the software Hadoop (JoshBaer, 2015) using these environments**. Firstly, the aim is the reduction of the Hadoop deployment cost using the cloud. Secondly, we discuss and analyze the assumptions learned from the HPC experiments and their applicability in the big data context. Thirdly, the Hadoop community finds an in-depth study of the resource consumption depending on the deployment environment. We come to the point that the use of the Docker container gives better performance in most experiments. Besides, the energy consumption varies according to the executed workload.

# 1 INTRODUCTION

The cloud-computing domain is based on the quality of the services offered to customers and the capacity of providers to ensure performances and security. The virtualization tools are the most important technology that has the capacity to hide the complexity of the infrastructure and to optimize resource exploitation. It helps providers to reduce costs. The virtualization technology was introduced in 1960 by IBM (Wen et al., 2012). It transparently enables time-sharing and resource-sharing on servers. It aims at improving the overall productivity by enabling many virtual machines to run on the same physical support. Many categories of virtualization tools (Wen et al., 2012) are used in data centers. In this paper, we classify them on the full and light virtualization: The full virtualization is based on the management of virtual machine (VM). The VM is guest operating system (OS) that runs in parallel over physical hosts. A hypervisor ensures the interpretation of instruction from the guest OS to the host OS. The light virtualization is based on the management of containers on a physical host, the containers share functions from the kernel of the host OS and have direct access to its library. In the last decade, the light technology of virtualization has been shifting quickly, it allows to obtain a cost-effective clusters of servers. Docker is the most sophisticated tool in its category; it offers a large and more intensive range of capability to manage hardware resources. This classification is also used in (Reshetova et al., 2014). Traditionally, the cloud computing and big data (Gandomi and Haider, 2015) environments are mainly based on the heavy virtualization tools. The main reason is the companies' lack of confidence in the following points: (*i*) the emerging technologies,(*ii*) the respectful efficiency of heavy virtualization,(*iii*) the complete isolation of the environment between guests and host OS. Nowadays, the Docker technology offers multiple capabilities of **resource isolation**. It reaches an adequate level of maturity and it can be tested with big data tools. Hadoop software is a big data environment, it is based on the MapReduce Model, which was introduced by Google in 2004 (Dean and Ghemawat, 2008) as a parallel and distributed computation model. It is largely adopted in companies and data centers; for example Facebook (JoshBaer, 2015) and Amazon (JoshBaer, 2015) use it to answer the computation needs.
In this work, we study and **compare the two cate-**

**gories of virtualization**. The experiments must be made using the Docker technology, the VMware technology and the Hadoop software. We therefore **analyze the influence of the platform's resource variation**. During the evaluation, we consider *(i)* the completion time of the workload,*(ii)* the quantity of hardware resources and *(iii)* the energy consumption criteria. We prove, then, that this technology gives a cost-effective cluster with a better efficiency in most cases.
The remainder is as follows. In section two, the previous studies on literature are presented. In section three, concept and terms used in this paper are reminded. In section four, the methodology used in the experiments is presented. In section five, the results are presented and discussed. The conclusion is presented in the last section.

# 2 RELATED WORKS

Since the last decade, Hadoop has been interesting the scientists community. Many benchmarks and evaluation tests have been done in order to evaluate its performances or to compare it to other softwares. There are mainly two levels in the benchmark of the software Hadoop.
The first one focuses on the comparison of Hadoop with the existing engine in the big data computing. For example, Fadika et al. (Fadika et al., 2012) benchmark Hadoop with three data-intensive operations to evaluate the impact of the file system, network and programming model on performances. Stonebraker et al. (Stonebraker et al., 2010) compare Mapreduce model to parallel database, they focus on the performance aspect. Pavlo et al. (Pavlo et al., 2009a) prove that Hadoop is slower than two state-of-the-art parallel database systems, in performing a variety of analytical tasks, by a factor of 3.1 to 6.5. Jiang et al. (Jiang et al., 2010) give an in-depth study of MapReduce performance to identify bottleneck factors that affect the performance on Hadoop, they show that the best tuning of these factors improves the same benchmark used in (Pavlo et al., 2009a) and (Pavlo et al., 2009b) by a factor of 2.5 to 3.5. Zechariah et al. (Fadika et al., 2011) compare Hadoop, LEMO-MR and twister (three implementations of the MapReduce model). Gu et al. (Gu and Grossman, 2009) compare Hadoop software (HDFS/ MapReduce) to the softwares Sector/Sphere. Jefrey et al. (Shafer et al., 2010) focus on the file system to identify bottlenecks, they identify the weaknesses of the Hadoop file system to solve and the best practices to follow in the cluster deployment.
The second one focuses on the performances and the energetic consumption of Hadoop using different deployment architectures. For example, Kontagora et al. (Kontagora and Gonzalez-Velez, 2010) benchmark Hadoop performances using full-virtualization (using VMware Workstation). The paper (Xu et al., 2012) evaluates Hadoop's performances using openStack, KVM and XEN. It compares performances using openStack deployment with the physical deployment. (Gomes Xavier et al., 2014) compare the Hadoop software using different tools of container technology, however, neither Docker technology nor heavy technology are considered in the comparison.
The Docker technology is benchmarked in other contexts as the HPC technology. For example, Xavier et al. (Xavier et al., 2013) present an in-depth performance evaluation of the containers based on the virtualization for HPC. They present the evaluation of the tradeoff between performance and isolation. In the same context, (Gantikow et al., 2015) compares the job executions using containers with executions using physical infrastructure deployment, it confirms that the overload due to the use of the container and the time completion are about 5 %. (Reshetova et al., 2014) analyzes the resource isolation in the context of Docker technology and confirms that container isolation is less secure than isolation offered by traditional tools of virtualization (heavy). For an accurate study, (Peinl and Holzschuher, 2015) presents the state of the art of all open source projects, which adapt Docker technology to the context of the Cloud.
We conclude that the Docker technology has been evaluated in the context of HPC technology, which has its specificity. In most cases, **the big data and HPC are two divergent fields of technologies**. Each one has its own scheduling policies, resources requirements workloads affinities. The topic of this paper focuses on the use of Hadoop software with the Docker technology as a light virtualization tool. It compares this emerging technology with the traditional virtualization technology. It focuses on the resources exploitation, the time completion of the benchmarks and the energetic consumption. It analyse and discuss assumptions acquired from experiments performed in the HPC context.

# 3 BACKGROUND

This section contains definitions of various concepts and terms used in this work. It presents the Hadoop software characteristics and it defines the heavy and light virtualization technologies.
Google introduced the model MapReduce as a

distributed and parallel Model for data intensive computing. Every job generates a set of "map" and "reduce" tasks, which is executed in a distributed fashion over a cluster of machines. "Map" tasks have to be executed before "reduce" tasks. Tasks have to be executed the nearest to the needed data input. Data outputs of tasks map are transferred from the machine where tasks "map" run to the machines where "reduce" tasks run using the network.

## 3.1 The Hadoop Implementation

Hadoop implements the MapReduce model; the computation level is named "Yarn" and is composed of three elements, which manage job execution. At first, the Resource Manager (RM) is the master daemon; it assures synchronization over different elements and distributes resources between jobs. On a second point, the Node Manager (NM) is the responsible for the resource exploitation per slave machine. The Application Master (AM) is responsible for managing the lifecycle of a job. The scheduler in the RM is responsible for the management of the resources.The scheduling policies are based on these assumptions:

1. the scheduler considers the homogeneity criteria of the cluster thus slave machines run jobs at the same rate.
2. the tasks progress linearly during a they tend to finish in waves, thus tasks having a low progress rate are considered as slow tasks
3. the tasks in the same category require the same amount of resources

The storage level is named Hadoop file system (DFS) and is composed of the NameNode (NN) as a server, which contains the cartography of blocks's file. The datanode is the second element of the storage architecture: it is responsible for maintaining data blocks and communicates with namenode to perform operations like adding, moving, deleting. It also applies a number of NN decisions like ensuring data replication and load balancing operations. The sizes of the files in DFS are from megabytes up to terabytes. They are partitioned into data blocks. The size of a block is a decisive point to reduce the duration of the workload execution. When the scheduler cannot assign tasks to machines where data are stored, network bandwidth is allocated to migrate blocks.

## 3.2 The Heavy Virtualization

The heavy virtualization consists in a virtual machine monitor and in a virtual machine (VM). VM has its own operating system that is completely isolated from the host operating system. The virtual machine concept is the basis of the full and para-virtualization approach (David, 2007). VMs have their own booked memory, disk space, network bandwidth and CPU's share. Thus we cannot afford to ignore the caused overhead, which is due to :*(i)* the virtual device drivers, *(ii)* the intermediate level which transforms instruction of the guest OS, *(iii)* the hypervisor that gives the administrator the possibility to run in parallel many OS per physical host. Either commercial or open source solution, a big work is done to limit mentioned overhead. The resource isolation in the full virtualization approach is at the hardware (Intel, 2015) and software level. We use VMware workstation® hypervisor to manage VMs in our experiment.

## 3.3 The Light or Container Technology

Either LXC and Docker containers use the kernel control groups (Cgroups), systemd (cores, 2015) and kernel namespaces libraries for (1) limiting and isolating resource consumption and (2) the process management.

In the context of this work, the Docker container guarantees the same function as the virtual machine and it has the same architecture. It is based on a management engine, it has the same role of the hypervisor in the traditional virtualization. However, the resources policy used for the containers management is more flexible than the one issued from the policy used in the full virtualization approach. CPU resources are an example, we can (1) fix the number of the CPU cores to allocate to each container or (2) define a relative share of the CPU resources between all containers on the physical host. In the second policy (2), the containers benefits from free CPU resources disposed on the physical host and releases them when they will be used by another process. Concerning the memory resources, a container requires consumed memory not provisioned memory, thus the containers offer better management of idle resources than VM.

The Docker technology introduces policies to manage four resources: memory, CPU, network IO and disk space management. Containers are able to share the same application libraries and the kernel of the host. The intermediate level that transforms instructions from guest to host OS is limited, therefore the container technology presents a lower overhead, it is considered as light virtualization. In big companies like Facebook and Yahoo, a cluster of Hadoop contains a large number of machines. The optimization of the resource exploitation offers the opportu-

Table 1: Configuration of machines (physical or virtual) used in the experiments

| | Host machine | Client machine |
|---|---|---|
| Processor | Intel ®Xeon(R) CPU E5-26200 @ 2.00GHz | |
| CPU cores | 12 | 2 cores (4 threads) |
| RAM (GB) | 31.5 | 5 |
| HDD (GB) | 500 | 80 |
| OS | Ubuntu 14.10 | |

nity to reduce costs and increase benefits. The companies profit from the virtualization in the cloud to improve resource exploitation. As the energy management presents an important field, much research over the Cloud aim minimize the electric consumption of the data center. This paper analyse the effect of the use of virtualization tools over the energetic consumption. It presents proportional relations between different kinds of resources and the consumption of energy. It is important to mention that the energetic gain over a cluster of four machines will be weak. The idea is to detect the variation of the consumption as small as it is. In a large cluster scale, the variation in energy consumption is not negligible and has an important impact on the overall cost.

# 4 Methodology

The experiments are repeated with both types of virtualization tools. The first topic of this work is to compare the performance variation using the two technologies of virtualization; we compare the time completion of the used benchmarks. The second topic is to focus on the Docker technology and give and in-depth study of this technology; we aim to identify the inadequate or badly spent resources. The third topic analyses the variation of the energy consumption according to the experiments and tests. Two sets of experiments have been carried out, the first set uses an homogenuous cluster of machine. In the second set, we vary resource capacities to thoroughly analyze performance variations between heterogeneous and homogeneous platforms. CPU, memory, hard disk and total load over physical machine are recuperated during the experimentations. We consider the time execution of the job. In all these works, the physical host has 12 CPU cores, 31.5 GB of RAM and 500 GB of hard disk (Table 1). The experiments are partitioned on two main parts: when we study Hadoop in homogenuous cluster, the virtual machines and the containers have the same configuration; they have 2 cores (and 2 thread per core), 5 GB of RAM and 80 GB of hard disk (Table 1). To ensure the best evaluation of the platform, some configuration parameter could be fixed. For example, the rate of data replication is two (this number is depending on the size of the cluster) and the capacity of node manager is set to 3 GB of RAM and 3 cores. When we address the problems in the heterogeneous cluster, we use another configuration of virtual machines (VMs and containers) depending on the resource we are studying. The experiments are based on two levels, the first one considers two slave machines and the second one considers four slave machines. The slave machines can be virtual machines or containers. All experiments are repeated 5 times. The Ganglia software is used to recuperate LOAD, CPU, RAM metrics. It overloads the Hadoop cluster with 2 per cent (Intel, 2015). The software hsflow is combined to Ganglia to retrieve I/O bound of the hard disk access on the slave machines. These metrics offer the possibility of an in-depth study in the variation in resource utilization during experiments. In order to measure energetic consumption, we use a specific engine mounted to the electrical outlet, it measures overall the energy consumption of the cluster machines every 2 seconds and save it on an external memory card. Four workloads (TestDFSIO-read, TestDFSIO-write, Teragen, Terasort ) are used in our benchmarks. In order to reach the topic of this work; we use the benchmarks Teragen and TeraSort and TestDFSIO. They are used by Vmware organisation; intel (Huang et al., 2010) and AMD (Devices, 2012) to evaluate their products. They are considered as a reference and are used in many other works like (Fadika et al., 2012). The first kind of workloads is Teragen and TestDFSIO. They stress the hard disk and I/O resources, they are based on a set of "map" tasks which writes random data in HDFS in the a sequential manner. In these works; they generate three sizes of data 10, 15 and 20 GB using 2 then 4 slave machines.

The second one is TeraSort, this benchmark stresses: memory, network and compute resources. Each data generated with Teragen is sorted with Terasort. Terasort is knwon for the capacity to aggregate output of the Teragen workload. It is based on a set of "map" tasks and "reduce" tasks. The job TeraSort is forced to use four reduce tasks, we aim to dispatch the compute on many slave machines. The four workloads (TestDFSIO-read, TestDFSIO-write, Teragen, Terasort ) used in the evaluation are based on the MapReduce model; each of them has the capacity to stress specific resource thus the evaluation results will be more accurate.

Hadoop is designed to work on a homogeneous clus-

Table 2: Configuration of Slave Machines and (NM) Used in Heterogenuous Context

| Resources | Slave machine 1 | Slave machines 2 and 3 | Slave node configuration |
|---|---|---|---|
| CPU (cores/Vcores) | 4 cores (4 threads) | 2 cores (4 threads) | 6 Virtual cores |
| Memory (GB) | 10 | 5 | 6 |
| HDD space (GB) | 80 | 80 | - |

ter. Defined policies (configuration of files and the default schedulers) don't consider the configuration of machines when they schedule tasks, however, the clusters and technologies grow up continuously and companies don't have guarantee to supply the cluster with the same machine's configurations. Thus, we study the influence of the variation of the machine configuration on the performance of the workloads executions. We vary the quantity of the resources of the Hadoop slave machines and we analyse experiments results. We double the RAM and CPU resources of slave machines in the cluster and the resource capacities of the slave node. The slave node is the Node Manager (NN) of the Hadoop's cluster. The table 2 introduces the configuration of the slave machines (VM or container) and nodes (which give the configuration of slaves in the Hadoop) considered at this part of experiences.

# 5 Experimental Results and Discussion

In this section, we provide the results of the experiments. The first subsection discusses the influence of the execution workloads on the performance of the two types of virtual clusters. In the second subsection, we focus on the variation in the resources i.e. CPU and I/O bounds. In the third subsection, we consider a heterogeneous cluster to analyse the variation of resource utilizations during experiments. In the fourth subsection, we study the influence of overload of the energy consumption.

## 5.1 Evaluation of the Machine's Overload Capacity

The overload of a machine can be defined as the difference between load of a physical machine (without any slave machine) and load after the start of slave machines on it. Figure 1 presents $(i)$ the overload of the physical machines without the running of any slave machines $(ii)$ the overload of the physical machine with slave VM when they are idle $(iii)$ the overload of the physical machine with Docker containers when they are idle. We have noticed that the virtual

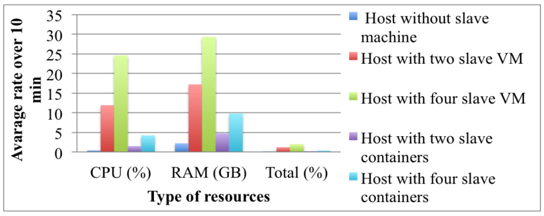



Figure 1: Resources overload with different number of virtual machines and different types of virtualization tools

machines reserve total configured memory since its start: thus 17 GB of memory is booked (3 VM) for cluster with two slaves and 28 GB is booked for the cluster with 4 slaves (5 VM). But the containers use resources only when they need them. The host uses 5 GB of memory with three containers and 10 GB with 5 containers. This is the minimum memory needed to start hosts, guest operating systems and Hadoop daemons. The Figure 1 shows that the overload is measured between 3-5% for Docker containers and between 10-25% for the commercial tools. During experiments, we record overload of the physical machine. Figure 3 compares the overload capacity using the two technologies of virtualization and the workload Teragen. These experiments also consider different size of Hadoop cluster and 20 GB of generated data. We remark in these conditions that container is lighter than traditional virtual machine and for axample, the workload Teragen causes less overhead than VM. The Figure 3(a) illustrates the total overload due to the execution of Teragen. The difference is interesting because it is a large difference in load between the two types of virtualization. The heavy virtualization is characterized by the reservation of the needed memory when these VMs start. It is visible from Figure 3(b) that the amount of memory reserved by the traditional virtualization increases compared to the results of memory consumption, shown in Figure 1. The additional memory is used by the hypervisors to interpret the guest operations which are executed by the host operating system.

Docker technology uses only the amount of memory needed to run their process, otherwise memory would be released. This behavior is due to the Docker container policy. The last requires consuming memory not a provisioned memory, thus the memory management in Docker is more flexible than the memory management in traditional virtualization tool. The Docker containers cause less memory overload than traditional VM which reserves 100-200 MB memory per VM for hypervisor. In addition, traditional VM independently reserves a fixed amount of memory. The Docker containers offer the possibility to fix a maximum amount of memory a container can use. However, when this memory is not explored by the container, it can be used by another processes. Concerning the variation of the CPU cores and memory, we present analyses and we give an in-depth description. We take as an example the execution of different job with 20 GB of data and we noticed that with the jobs Teragen and Terasort, the cluster using Docker technology is more efficient than cluster with traditional virtualization (Figure 2(a) and 2(b)).

We obtained same results for the same job, executed on the same size of a clusters. The use of two slave machines gives better performances than the use of four slave machines. Thus, the number of slave machines should be correctly chosen to avoid the degradation in performances. We give in next part an in-depth study of Hard disk and CPU bounds exploration.

## 5.2 I/O-bound Variation and CPU Bound

The TestDFSIO benchmarks are used to evaluate the HDFS health, they utilize the hard disk resource more than other resources as memory or CPU cores. We run TestDFSIO benchmarks on 2 and 4 hadoop slave machines, using the two technologies of virtualization. Then we also vary the writable data sizes (10, 15 and 20 GB). We present the experimental results of the disk write and read throughput and average IO in Figure 4. The results of the job execution: TestDFSIO-write, (Figure 4(a) to 4(c)) proves that the throughput and average I/O are inversely proportional with the overload measured during the execution. For example, using 4 VMs over 20 GB of data, the overload is about 85%, but throughput and averageI/O highly decreases. The management of hard disk bound influences directly the completion time of the workload execution. The results proves that Docker technology use a best policy to manage access to the hard disk compare to the VMware tool.

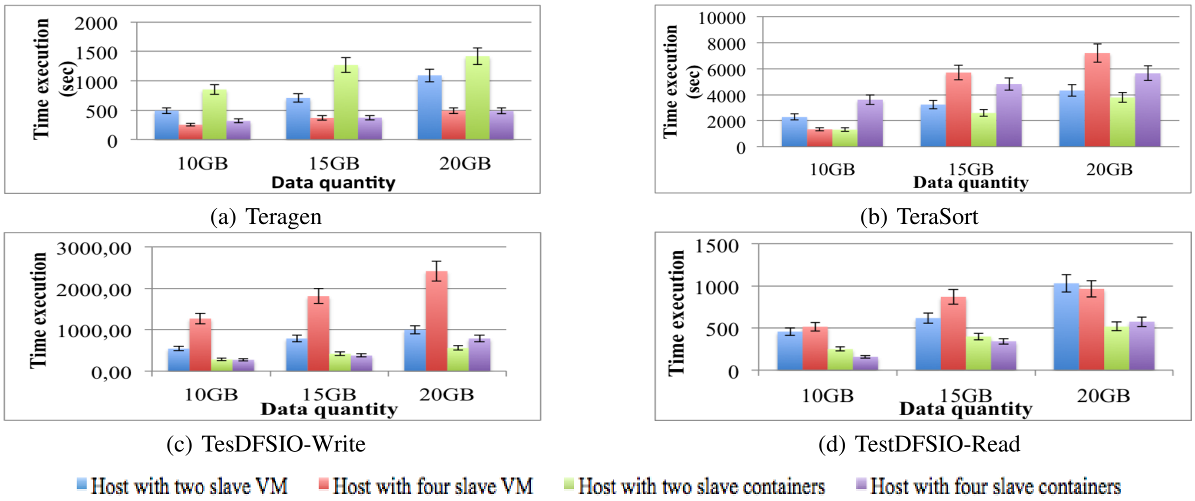
We use the workloads Teragen and Terasort to stress the CPU bound. It has a considerable influence over the performance and the energy consumption. In our experiments, we use two cores per slave machines. Docker technology offers two policies to manage CPU resources. The first method is to reserve a specified number of cores per container. The second one uses a relative share rate between containers. It associates a weight to each container and it shares the existing compute resources between them. Please note that all the experiments, explained earlier use the first reservation policy. We will focus on the fair share policy in Section 5.3. We notice that using Teragen (Figure 3(a)) and TeraSort; containers cause about the half of the CPU overload than traditional virtualization tool. Figures 2(a) and 2(b) shows the completion time of the used workflow over a cluster of 2 and 4 slave machines. The cluster with two slave machines gives better performances than the cluster with four. One reason is the architecture, the use of four slave machines increases the competition to access resources and the total overload increases in consequence. For example, when we evaluate the cluster with two slave machines, six cores (CPU) are booked for the virtual cluster so the host OS has the six other cores to use and to run the instructions. However, cluster with four slave machines uses 10 cores (CPU) thus only two cores are used by the host OS. We observe the same behavior for the memory resource use. Thus, we note a performance degradation. As we use the same policy to manage CPU resource in the two cases of study ( 2 and 4 slave machines), we conclude that the main reason of the performance degradation is the management of throughput and memory policies. We use TestDFSIO read workload to test the read throughput. The results are summarized on Figure 4(a) to 4(c). We noticed that the running of multiple slave machines on the same physical host creates a concurrent access to the hard disk. Despite the replication of data used in Hadoop (which is equal to 2), there is a difference in performance between slave machines, depending on the used technology. We give an in-depth description of the memory management policies in Docker technology in section 5.1.

Despite the congestion of resources, when we work with a cluster of four slave machines, in most cases, the Docker container offers better performances in most cases.

In the next subsection, we focus on the execution of the workloads on a heterogeneous cluster and compare the two technologies of CPU management, available in the Docker technology.

## 5.3 Performance Variation in a Heterogeneous Cluster

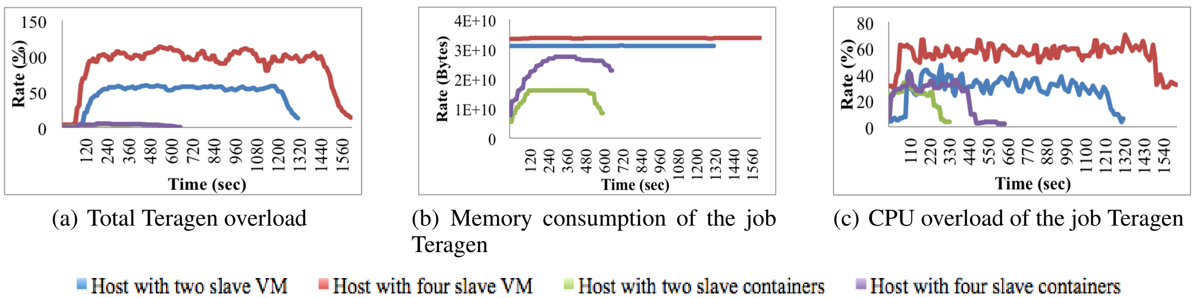
We analyse in this subsection the influence of the heterogeneous cluster on the performance of Hadoop. The first step of the experiments uses two different configurations of slave machines. The second step




(a) Teragen (b) TeraSort

(c) TesDFSIO-Write (d) TestDFSIO-Read

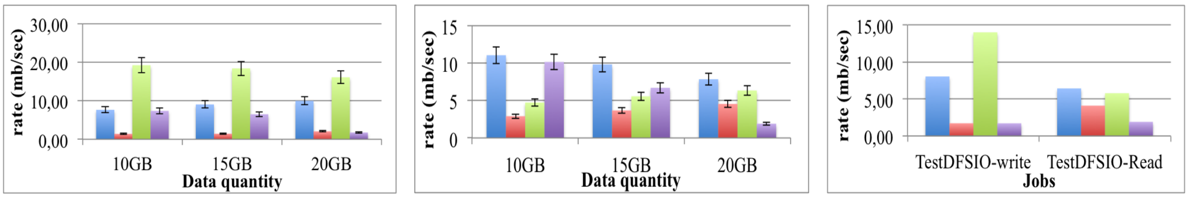
Figure 2: Time execution in function of the data quantity and type of slave machines




(a) Total Teragen overload (b) Memory consumption of the job Teragen (c) CPU overload of the job Teragen

Figure 3: An exemple of the variation of the overload due to use of different virtualization tools

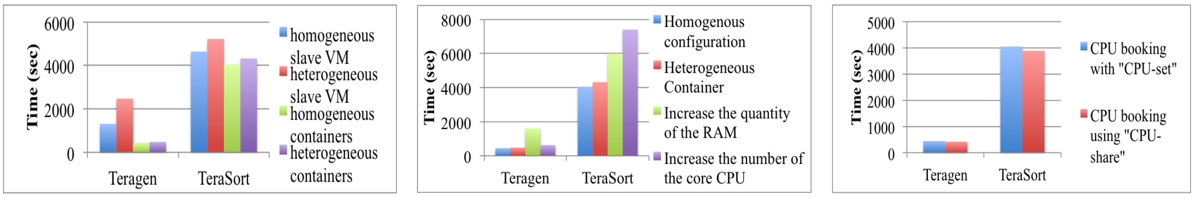
changes the capacity of each resource and analyses the influence of each resource variation in Hadoop performances. In the first step, we use an additional configuration of slave machine, then we obtain new Hadoop cluster with two types of slave machine's configurations: one configuration has 4 cores, 10 GB of RAM and 80 GB of hard disk, the second has 2 cores, 5 GB of RAM and 80 GB of hard disk. The results of the execution of the workloads Teragen and Terasort over a heterogeneous and homogeneous cluster of three slave machines is presented in Figure 5(a). It confirms that Hadoop outperforms better with homogeneous cluster. In the Figure 5-b, we focus on the Docker technology performances. We run the two jobs in this four use cases: (1) homogeneous cluster of three machines, (2) heterogeneous cluster (described in table 2), (3) heterogeneous cluster with increasing the memory and (4) heterogeneous cluster with increasing only of the number of cores. Then we conclude from this experiment that varying only RAM or CPU resources is not helpful for the Hadoop performances. There are two reasons of this conclusion. The first one is that increasing the capacity of memory in client machines stresses the host operating system and limits its performances. In the same manner, increasing the number of virtual cores per container limits the number of CPU cores used by the host system and decreases its computing capacity. The second one is noted at the scheduling level. After observing the tasks assignement at the Hadoop scheduler level with the two technologies, in the first third of the time execution, we noticed that workloads have a high rate of tasks failures on the slave machine (SM) two and three. However, there is no task failure in the first SM. These results are due to: *(i)* The homogeneity of the cluster, considered by the scheduler (capacity scheduler is used in experiments). *(ii)* The mismatch between the resource definition in the configuration files. *(iii)* The available resources on the cluster. During the remaining period of execution of Terasoft and Teragen, the scheduler has the tendency to re-run the failed tasks with double capacity of resources. The scheduler adapts its behavior and affects the major quantity of tasks, which have double capacity of resources to SM-1. For example, behind the capacity of Nodemanager's resources in table 2 (slave daemon on




(a) TestDFSIO-Write Average IO (b) TestDFSIO-Read Avarage IO (c) Write/Read Throughputs (20 GB)



Figure 4: Throughput and Average IO of the job TestDFSIO execution with different virtualization tools




(a) Influence of the heterogeneous cluster (two types of virtualization) (b) Influence of resources variations of Docker containers on the execution of jobs (c) Comparison between the two policies for managing the CPU resource allowed by the Docker technology

Figure 5: Completion time of the workloads Teragen and TeraSort with heterogeneous platforms

the Hadoop cluster), the SM-1 runs three tasks all the time, two tasks have 2GB of memory and the third has 1GB of memory. The SM-2 and SM-3 have the tendency to execute tasks with 1GB of memory, four times more than tasks having 2 GB of RAM. We notice that the scheduler always affects one Vcores (virtual cores) per task. This is caused by the fact that Hadoop considers the criteria for resource homogeneity during scheduling of the tasks.
In Figure 5(c); we focus on the CPU resources. In previous experiments, we use the reservation policy to affect CPU cores in containers (section 4-3). On the next step, we compare the two policies, given by Docker to explore computing capacity between containers. The results show that there is a thin difference between policy on the described environment of experiments. The sharing method (with CPU-share option) performs better than the affectation method (reservation policy with the CPU-set). The share policy gives the opportunity to share unused compute resources between containers and don't limit them to a specific number of cores. As the Hadoop context is concerned, The share policy increases the capacity of slots on the slave machines. Thus, it increases performances without having the negative aspect on the host OS. When the number of slave machines per physical host is maintained, the share policy gives better performances. It ensure a minimum rate of computational capacity per container. When the host machine has a free computational resources, these resources are shared respecting the relative share between containers. As a result, the compute capacity per hadoop NM deamon increases and will have a good influence on the completion time of the workloads. When the overload limit is reached, the two methods have the same behavior and the decrease on performances.

## 5.4 Evaluation of the Energy Consumption

The energy consumption is an important issue in the big data context i.e. Yahoo deploys a Hadoop cluster over more than 2000 servers; Facebook deploys Hadoop over 600 servers; General Electric deploys Hadoop on a cluster of 1700 servers. At this scale, the energy is a critical aspect which influences considerably the cost of cluster exploitation. A research realized by the U.S. Environmental Protection Agency (Agency, 2007) and the Natural Resources Defense Council (Council, 2014) announced in 2007 that the cost of energy consumption for cluster management was very high. The commission in the European Union defined the code of conduct on datacenter energy efficiency since 2008 (for Energy and (IET), 2015). Through the experiences, it is clear that the load and the energy consumption are proportional, when we run 4 slave machines. The overload and energy consumption increase and they are higher than the case of 2 slave machines. We can conclude that when the overload of physical machine increases (more than 85 %), the performance degrades and then, energy consumption increases. Installing many vir-

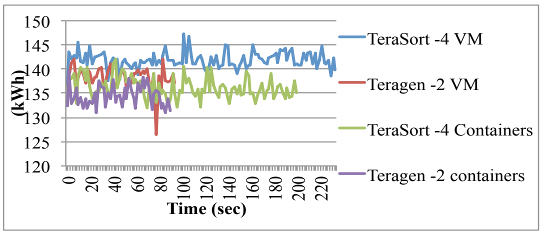



Figure 6: Energetic Consumption of different size of clusters with jobs Teragen and Terasort

tual machines on the physical host increases the energy consumption and they have negative influence on the job's execution performances. The overload on physical host is proportional to the number of slave machines and the workload running on them. Figure 2(a) and 2(b) show the completion time of Teragen and TeraSort workloads over a cluster with different slave machines. The cluster of two slave machines is better performing than the cluster with four machines and has a bit lower consumption than four slave machines cluster. The virtualization technology is used by the server providers to manage the load on the physical machine and to optimize energetic consumption. The overload on the physical machine is the aggregation of all overload of their guest when the VMs run in a higher load. Working with the same type of job, size of cluster and quantity of data (Figure 6), there is a thin difference between the use of the two virtualization tools. Docker technology consumes less energy than traditional tools. This one is caused by the use of containers instead of the overload due to the use of virtual machines.

# 6 CONCLUSIONS

This paper has four objectives: *(i)* The analysis and the study of many assumptions concerning the configurations of big data platforms. *(ii)* The comparison of performances of the Hadoop platform with the two technologies of virtualization. *(iii)* The study of the variation of the performance for the case of homogeneous and heterogeneous platforms and *(iv)* The deals with the energy consumption on the Hadoop cluster. (refer to Section 5.2). We can confirm that assumptions performed from experiments in the HPC domain and focus on the container technology. The deployment of the Hadoop cluster either by using traditional virtualization or containers technology, optimizes the resource exploitation and minimizes idle resources. However, using the two technologies decreases the efficiency and the cluster's performance. In general, the container technology exceeds the traditional virtualization technology. In the major part of the test, the containers cause less overhead on CPU resources, however the two policies given by the Docker technology in order to manage computing resources should be used carefully. Hadoop is based on the sharing of the computing capacity between a numbers of slots through time. The fair share policy can increase the rate of computing resources. However, it strongly influences the performance of the host operating system since it limits the resources of the clusters. We choose to fix the number of cores per slave machine. Hence, this method offers an accurate report about the resource exploitation and it allows a better comparison between these results.

The containers have an efficient policy to manage memory resources; free memory can be recuperated by the host operating system in order to improve general performance of the physical host. The Hardware and network bandwidth are shared between guests that are localized on the host. We only consider the resource isolation, the other kinds of isolation (like user or session isolation) are not targeted in this work. The two technologies used in this paper can isolate CPU, memory and Hard disk resources. However, despite the evolution of the hard disk resource isolation (as blkio controller), the access rate to hard disks remains an open problem. The main reasons are: *(i)* the overload due to the workload execution or due to the number of slave machines per host. *(ii)* The I/O scheduling, when tasks are running; a big quantity of data is transferred between the slave machines. It ensures data replication and merging between tasks. Thus, the I/O scheduling has a direct influence on the Hadoop cluster's efficiency. The I/O environment considers the network bandwidth and harddisk access.

The Hadoop software is adapted to be used with homogeneous platforms. However, hardware technologies are changing continuously and it is not possible to ensure the same Hardware configurations when the cluster is evolving. On the other hand, our experiments argue that heterogeneous platforms degrade the performances, the main reason is being the scheduling policies because the scheduler in Hadoop is performed to work on homogenuous cluster and the only policy used to speed up the processing of the applications is to run a copy of delayed tasks on other machines. The energy consumption is directly related to the load of resources on the physical host i.e. the higher is the load of physical host, the higher is the energy consumption. However, the performance depends on the number of slave machines per host and it also depends on the execution of workloads.

In this work, benchmarks argue that the light virtual technology is the best to use in the Hadoop context.

In the future, we will focus on the optimization of the Hadoop performances by working on the scheduling policies, in order to improve performances. The approach mentioned in (Jlassi et al., 2015), presents the definition of the scheduling problem on the Hadoop cluster. We take into account the performances and the energy consumption in a bicriteria problem.

## ACKNOWLEDGEMENTS

This work was sponsored in part by the CYRES GROUP in France and French National Research Agency under the grant CIFRE n$^{o}$ 2012/1403.

## REFERENCES

Agency, U. E. P. (2007). Report to congress on server and data center energy efficiency.

cores (2015). Getting started with systemd. `https://coreos.com/docs/launching-containers/launching/getting-started-with-systemd/`.

Council, N. R. D. (2014). American data centers are wasting huge amounts of energy.

David, M. (2007). Understanding full virtualization, paravirtualization and hardware assist. White Paper.

Dean, J. and Ghemawat, S. (2008). Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113.

Devices, A. M. (2012). Hadoop performance tuning guide - amd.

Fadika, Z., Dede, E., Govindaraju, M., and Ramakrishnan, L. (2011). Benchmarking mapreduce implementations for application usage scenarios. In *Grid 2011*, pages 90–97.

Fadika, Z., Govindaraju, M., Canon, R., and Ramakrishnan, L. (2012). Evaluating hadoop for data-intensive scientific operations. *IEEE CLOUD 2012*.

for Energy, I. and (IET), T. (2015). Data centres energy efficiency. `http://iet.jrc.ec.europa.eu/energyefficiency/ict-codes-conduct/data-centres-energy-efficiency`.

Gandomi, A. and Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2):137 – 144.

Gantikow, H., Klingberg, S., and Reich, C. (2015). Container-based virtualization for hpc. In *CLOSER 2015*, pages 543–551.

Gomes Xavier, M., Veiga Neves, M., and Fonticielha de Rose, C. (2014). A performance comparison of container-based virtualization systems for mapreduce clusters. In *22nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*.

Gu, Y. and Grossman, R. L. (2009). Lessons learned from a year's worth of benchmarks of large data clouds. MTAGS '09, pages 31–36. ACM.

Huang, S., Huang, J., Dai, J., Xie, T., and Huang, B. (2010). The hibench benchmark suite: Characterization of the mapreduce-based data analysis. In *Data Engineering Workshops (ICDEW), 2010*, pages 41–51.

Intel (2015). Intel virtualization technology (intel vt). `http://www.intel.com/content/www/us/en/virtualization/virtualization-technology/intel-virtualization-technology.html`.

Jiang, D., Ooi, B. C., Shi, L., and Wu, S. (2010). The performance of mapreduce: An in-depth study. *Proc. VLDB Endow.*, 3(1-2):472–483.

Jlassi, A., Martineau, P., and Tkindt, V. (2015). Offline scheduling of map and reduce tasks on hadoop systems. In *CLOSER 2015*, pages 178–185.

JoshBaer (2015). Hadoop wiki powerby. `https://wiki.apache.org/hadoop/PoweredBy`.

Kontagora, M. and Gonzalez-Velez, H. (2010). Benchmarking a mapreduce environment on a full virtualisation platform. In *CISIS 2010*, pages 433–438.

Pavlo, A., Paulson, E., Rasin, A., Abadi, D. J., DeWitt, D. J., Madden, S., and Stonebraker, M. (2009a). A comparison of approaches to large-scale data analysis. SIGMOD '09, pages 165–178, New York, NY, USA.

Pavlo, A., Paulson, E., Rasin, A., Abadi, D. J., DeWitt, D. J., Madden, S., and Stonebraker, M. (2009b). A comparison of approaches to large-scale data analysis. SIGMOD '09, pages 165–178.

Peinl, R. and Holzschuher, F. (2015). The docker ecosystem needs consolidation. In *CLOSER 2015*, pages 535–542.

Reshetova, E., Karhunen, J., Nyman, T., and Asokan, N. (2014). Security of OS-level virtualization technologies: Technical report. *ArXiv e-prints*.

Shafer, J., Rixner, S., and Cox, A. (2010). The hadoop distributed filesystem: Balancing portability and performance. In *ISPASS 2010*, pages 122–133.

Stonebraker, M., Abadi, D., DeWitt, D. J., Madden, S., Paulson, E., Pavlo, A., and Rasin, A. (2010). Mapreduce and parallel dbmss: Friends or foes? *Commun. ACM*, 53(1):64–71.

Wen, Y., Zhao, J., Zhao, G., Chen, H., and Wang, D. (2012). A survey of virtualization technologies focusing on untrusted code execution. In *IMIS'12*, pages 378–383.

Xavier, M., Neves, M., Rossi, F., Ferreto, T., Lange, T., and De Rose, C. (2013). Performance evaluation of container-based virtualization for high performance computing environments. In *PDP, 2013*, pages 233–240.

Xu, G., Xu, F., and Ma, H. (2012). Deploying and researching hadoop in virtual machines. In *ICAL '12*, pages 395–399.